Motion Filtering with Dynamics

Victor B. Zordan

College of Computing and Graphics, Visualization, and Usability Center
Georgia Institute of Technology
Atlanta, GA 30332-0280
victor@cc.gatech.edu

August 19, 1998

Abstract

This paper describes a technique for filtering and modifying human data using dynamic constraints and simulation. A simple control system calculates torques to track human motion data for a dynamic simulation of an articulated figure. The simulation generates smooth, physically plausible motion that maintains characteristics of the original. This technique is used to filter data for a variety of upper-body motions, animating two models with differing kinematic parameters and degrees of freedom. In addition, the system creates transitions from one motion sequence to another in the spirit of previous techniques. Further, the system imposes task and environmental constraints to generate believable behaviors and dynamic contacts.

Keywords

Computer animation, human figure animation, motion capture, dynamic simulation.

I. Introduction

As motion capture systems become more available and motion capture animation gains popularity, the demand for robust and flexible techniques for handling motion data increases. Human motion data recorded from commercial systems is rich with detail and realism, but can suffer from such problems as environmental noise, slipping markers, and spurious samples caused by errors in the capture process. Data is often played through kinematic skeletons with fixed limb lengths and idealized joints. With these approximations, errors accumulate in end effector positioning, resulting in poor interaction between a hand and its environment. Several semi-automatic techniques solve subsets of these problems but a great deal of motion editing is still done by hand due to a lack of suitable general techniques.

This paper presents a system which uses a dynamic model to filter motion data. Most existing techniques deal with motion filtering by fitting a kinematic model. Forward kinematics is used to fit motion to articulated characters by imposing limb lengths and joints constraints. In this fashion, individual marker data are transformed into a linked chain of bodies. Inverse kinematics (IK) is used to further modify motion data to achieve constraints such as end effector positions and joint limits in more sophisticated systems [4, 2]. However, the motion resulting from these kinematic techniques may satisfy the desired constraints and still appear physically implausible because the kinematic constraints do not capture dynamic characteristics such as smooth hand trajectories. Dynamic models produce smooth motion due to inertial properties. Further, dynamic constraints, such as complex impacts between a hand and its environment, are impossible to achieve with simple kinematic models, making dynamic models more desirable than kinematic models.

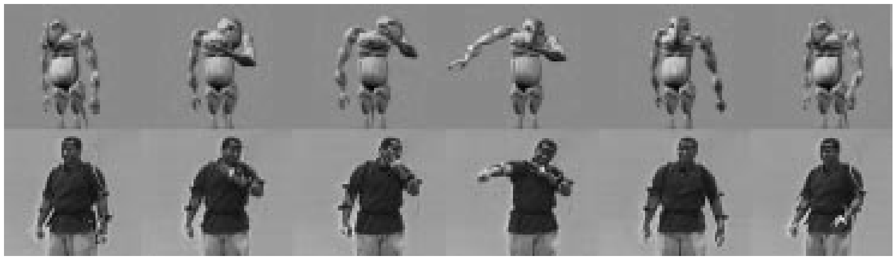

Fig. 1. **Motion Comparison - Dynamic Filtering vs. Live Motion.** Dynamic filtering produces motion that maintains the overall characteristics of the original but imposes the dynamics of the character being animated. The resulting motion has desirable qualities such as smooth hand trajectories.

This paper describes a simple but effective technique for filtering an existing motion sequence using a dynamic model. The technique uses raw human data as input to a tracking controller for a dynamic simulation of an articulated figure. The simulation is generated with the dynamic parameters of the figure to be animated and thus the filtering process creates motion with physical qualities specific to that character. The filtering system smoothes the input data, ignores outliers due to capture errors and small disturbances like those caused by marker slippage. The resulting motion is physically plausible and maintains overall characteristics of the original motion (Figure 1). Although the input data is not matched precisely, the filtered motion has desirable characteristics such as smooth hand trajectories and may be modified while maintaining dynamic realism.

In addition to basic filtering, the system described is able to generate smooth transitions between motion sequences. Motion sequences are often captured in small basis segments to be strung together as a post-process. However, a system for arbitrary transitioning is difficult because transitions between sequences are highly specified but under-constrained. A dynamic model is used to create physically plausible transitions and smooth the entire blended sequence by maintaining a consistent set of dynamic constraints for the duration of the motion. The dynamic filtering system affords a natural mapping for dynamic behavior control and dynamic environment interaction. A task controller is added to allow direct specification of motor-level behaviors in the editing process, resulting in dynamically plausible motions. In addition, the system is modified to impose dynamic environmental constraints such as hand collisions. Collision reactions are calculated and applied as forces, resulting in motion with realistic impacts. The motion sequence of a drumming behavior is modified so that the hand hits the drum in a believable way. A dynamic model can include the reaction forces required to satisfy dynamic constraints automatically. Example implementations are presented including details on modifications to the base-level system.

This paper describes a simple implementation of a dynamic model that filters raw orientation data for upper-body motions. The resulting motion is compared to the Cartesian trajectories of the original hand marker and the hand trajectory for a forward kinematic play back of the raw orientation data. The system is used to filter motions, to create smooth dynamically plausible transitions between captured sequences, and to modify motion by imposing dynamic constraints. The remainder of this paper is broken down as follows. The next section includes a review of the relevant background literature. The dynamic simulation and basic filtering system are described next. Transitions and dynamic constraints with example implementations follow. Finally, the last section concludes with results and discussion.

II. Related Work in Motion Editing

Motion capture technology has received much attention in recent years. Topics span the entire motion capture pipeline from performing a successful capture session [2] to interactive systems for motion editing [3, 4]. This research focuses on cleaning raw data and providing high-level control in the editing process using a dynamic model. In this section, relevant work in these areas is highlighted.

This work draws on previous methods to create and control a dynamic model. Although most techniques for cleaning raw data include a kinematic model, techniques for using dynamic models have been largely unexplored. However, the dynamics community has introduced numerous methods for creating and controlling dynamic systems. A number of hand-tuned simulation systems for animating rigid-body human-like characters have been introduced [9, 1, 7]. Automatic techniques for generating motion with dynamic systems have also been introduced [19, 12, 15]. A natural extension of these techniques is the use of dynamic simulations for modifying motion data.

High-level editing relies on modifying existing data by adapting motion segments to new situations and seamlessly reordering segments. Adapting segments has been approached by either modifying an existing motion with constraint solvers that fit user-specified changes [5, 20] or by generalizing sample behaviors to create controllable, parameterized motions. Systems that support generalizing behaviors vary in terms of underlying models but all attempt to generate believable, parametric motion from samples of data. Unuma, Anjyo and Takeuchi use Fourier interpolation to generate ranges of emotional walking and running gaits [14]. Wiley and Hahn use a tri-linear interpolation pyramid with time scaling to generate a generalized pointing behavior and walking gait for sloping terrain [18]. Rose, Bodenheimer and Cohen introduce a system that uses radial basis functions for generalizing a variety of behaviors including walking and running on varying terrain [10]. Their work also emphasizes parametric control of emotional expressiveness for base behaviors.

Demand for the seamless reordering of segments has lead to an emphasis on generating reasonable motion for transitions between capture sequences. Transitioning from one sequence to another in a realistic fashion is especially important in motion capture editing because motion is often recorded in small basis segments. Witkin and Popovic suggest a method for transitioning by blending from one sequence to another [20]. Rose and his colleagues present a more complex system for transitioning that uses an inverse dynamics formulation to constrain the generated motion [11]. They use a solver to minimize the torque expended during a transition according to the inverse dynamics model. They also suggest a simpler technique for cyclification, or transitioning from a sequence onto itself. Their work in inverse dynamics is most closely related to the efforts addressed in this paper by considering dynamics to modify motion data. However, this work introduces a technique for using a forward dynamic model to clean data and edit segments with dynamic constraints.

III. Dynamic Simulation of a Humanoid Upper Body

The basic filtering technique uses a dynamic simulation of a humanoid upper body and a simple tracking controller. Upper body motions are chosen as a testbed because the dynamic simulation is stable and easy to control.

Two upper body dynamic simulations are used consisting of rigid-body upper bodies, with static graphical legs. Eight to nine rigid links are connected with revolute joints of three degrees of freedom (DOFs), totaling 24 - 27 DOFs. The dynamic model was created by methods described by Hodgins and her colleagues [7] . Mass and moment-of-inertia information is generated from the graphical body parts and estimated density values. The equations of motion are calculated using a commercial solver, SD/Fast [13].

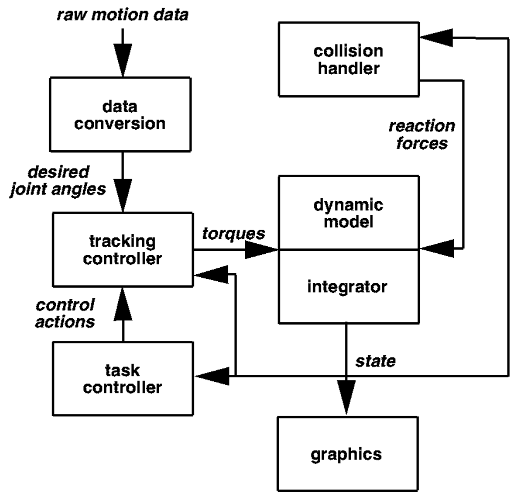

Fig. 2. **Dynamic Filtering System Layout.** Raw motion data is converted into smooth joint angles and input to a simple tracking controller. This controller calculates torques for the dynamic model which is integrated to generate motion. A task controller and collision handler may be added to achieve more complex dynamic constraints.

The dynamic simulations have characteristics of the underlying animated figure. That is, the dynamic parameters fit the mass and inertial properties of the character to be animated. So, the motion generated by the simulation is plausible for the particular character. Bodenheimer and his colleagues suggest methods for extracting values for skeleton parameters that match the human captured in the motion [2], but in general a character with different size and shape may be chosen for animation. Modifying motion with such a dynamic model accounts for differences caused by body scale. Other aspects of body scale as it apples to dynamic simulations are discussed by Hodgins and Pollard [6].

The dynamic simulations described do not detect inter-body collisions and have no notion of joint limits. These constraints must be imposed by the underlying motion and the tracking controller described next.

IV. Filtering with a Tracking Controller

Basic dynamic filtering uses the simulation described above and a simple tracking controller which calculates joint-space torques based on human motion data. The system diagram (Figure 2) shows the general layout and flow of information in this dynamic filtering system. Human data in the form of joint angles are input to the system, the controller determines torques from tracking errors and the simulation is integrated. A collision handler and task-level controller may be added as described later.

The tracking controller uses feedback and the human data to calculate appropriate joint-space torques to be applied to the dynamic simulation. Joint-space controllers with torque actuators have been used successfully to generate a variety of dynamic behaviors [9, 7, 16]. Like many of these systems, control torques are calculated using a proportional derivative servo (PD-servo) at each joint:

$$\tau = k_d\ (\dot{\theta}_{desired} - \dot{\theta}_{actual}) + k\ (\theta_{desired} - \theta_{actual})$$

where $\dot{\theta}_{actual}$ and $\dot{\theta}_{desired}$ correspond to the actual and desired joint velocities, and θ_{actual} and $\theta_{desired}$ correspond to the actual and desired joint angles.

The desired values for each joint at each time step are derived from the input motion data.

Fig. 3. **Filtered motion for an animated crowd scene.** A crowd of lively aliens are animated by filtering human motion data. For a variety of behaviors, the resulting motion is physically plausible and retains characteristics of the original.

Fig. 4. **Gesture motions for MC character.** The MC character gestures based on human motion data. Variability in characters such as different kinematic parameters and degrees of freedom affect the underlying dynamic simulation and generated motion.

Joint angle trajectories for $\theta_{desired}(t)$ are calculated from raw orientation matrices for individual body markers and hierarchical skeletal information as:

$$\Theta_{desired} = \Theta_{ii}{}^{\top}\Theta_{io}$$

where Θ_{io} and Θ_{ii} are the orientation matrices of the outboard and inboard bodies at joint i for a particular sample time. Joint angle data is converted to Euler angles and interpolated using Hermite splines to create a set of continuous trajectories. The resulting joint trajectories are then used by the PD-servo as target values. In this manner, the dynamic simulation is controlled to track the joint data.

Gain and damping terms are initially set to a nominal value by hand and then a simple search is employed. Gain values are chosen to minimize the error, ϵ, over an entire trajectory as:

$$\epsilon = \sum ||\theta_{desired} - \theta_{actual}||$$

A simple gradient is employed to solve this search resulting in motion that more closely matches the input data. More sophisticated search algorithms would undoubtedly find better solutions as would more complex error metrics such as an error calculated from the Cartesian position data from the original motion sequence.

This basic filtering system was used to generate an array of behaviors such as those in Figure 3 and Figure 4. The two characters have very different dynamic parameters but the same basic filtering system was used to generate motion that is plausible for their individual dynamics. Some simple modifications can be made to this base-level system to generate more interesting and complex motion. Two examples, transitioning and satisfying dynamic constraints are described below.

V. Plausible Transitions between Motion Sequences with Dynamics

Smooth transitions between basis motion sequences are required to maintain realistic, continuous motion. Often data is captured in segments corresponding to sets of basis behaviors. Transitioning a character from one basis behavior to another with continuous movement may be done in a number of ways with varying levels of flexibility and believability. One simple approach, used in commercial electronic games, requires choosing a particular *home* position and having each basis behavior begin and return to this *home*. This solution is robust but leads to restricted and repetitive transitions. More general systems for arbitrary transitioning are desirable.

Considering that the human motion underlying these basis sequences is dynamic, a dynamic model for transitioning seems appealing. A dynamic model enforces a consistent set of constraints for the duration of the motion. In this fashion, a transition will be physically plausible according to the dynamics of the character and will match the leading and following motion sequences if the filtering system is used. The dynamic simulation that is used to filter the basis behaviors generates a transition with the same qualities as the leading and following behaviors.

The filtering system is modified to support smooth transitions in the spirit of previous techniques. Transitions are performed by interpolating the desired angles from one basis motion to the next. As suggested by Witkin and Popovic [20], the technique implemented blends between two sequences of joint angles, θ_a(t) and θ_b(t). A new sequence of joint angles, θ_t(t), to transition from θ_a to θ_b is created by interpolating between the joint trajectories as follows:

$$\theta_t(t) = \theta_a(t)(1 - \omega(t)) + \theta_b(t)\omega(t),\ 0 \leq \omega(t) \leq 1,\ t_0 \leq t \leq t_1$$

The weight, $\omega(t)$, uses a simple ease in/ease out weighting scheme and times t_0 and t_1 correspond to the beginning and end of the transition. The interpolation used is a quaternion slerp algorithm similar to the one described by Watt and Watt [17]. The quaternions are then converted to Euler angles and the resulting joint trajectories are used as the desired values for the filtering system. By interpolating in quaterion space, the system may perform transitions without need for the Euler angle reformulation described by Bodenheimer and his colleagues [2]. For each transition, an animator specifies parameters for the time and duration of the sequence.

The resulting motion is smooth and physically plausible. Figure 5 shows a graph of hand positions for a transition of the MC character from a crowd-greeting gesture to a signaling gesture with the original filtered sequences shown. The slerp interpolation generates smooth desired joint trajectories and the inertial properties of the dynamic simulation further smooth and add disturbances that mimic how a physical character may perform this transition. The dynamics of the character remain the same throughout the motion and the result is smooth. Of course, many transitions will lead to inter-body penetration and unnatural postures and so, much of the skill in creating the transitions still relies in the animator's choice of parameters.

VI. Dynamic Constraints for Task and Environmental Interaction

Some complex constraints can be satisfied directly by a dynamic model. Dynamic constraints include specification of forces and/or motor actuation that is not easily achieved by kinematic models. Two types of dynamic constraints include motor-level task specification and highly dynamic environmental interaction. To understand each of these classes in the context of a dynamic model, we consider two illustrative example behaviors: a staff-pumping riot motion as a task constraint problem and a tribal drumming motion as a dynamic environmental constraint problem.

A dynamic model can satisfy motor-level task constraints where kinematic models break down. In some task-level constraints, the kinematics can not easily capture the complexity of the situation. Consider the simple behavior of a rioting character pumping a staff. When the raw motion data is played through a kinematic skeleton, errors and noise cause the staff to flail wildly with highly unnatural accelerations. The hand and staff may be constrained kinematically but this can easily result in motion where the staff appears massless. By using a simple controller in the dynamic system, the simulation yields motion for the staff that is more believable.

Implementation of this staff-pumping task requires little modification to the base-level system. A simple PD-servo in the wrist uses feedback to maintain an upright orientation for the hand and staff. In this manner, the wrist controller attempts to keep the staff upright while the rest of the body moves according to the original motion. The PD-servo gains for the wrist are hand-tuned so that disturbances are minimized and the staff sways in a realistic way under its own mass (see Figure 8). Other examples that combine motor-level control with the tracking

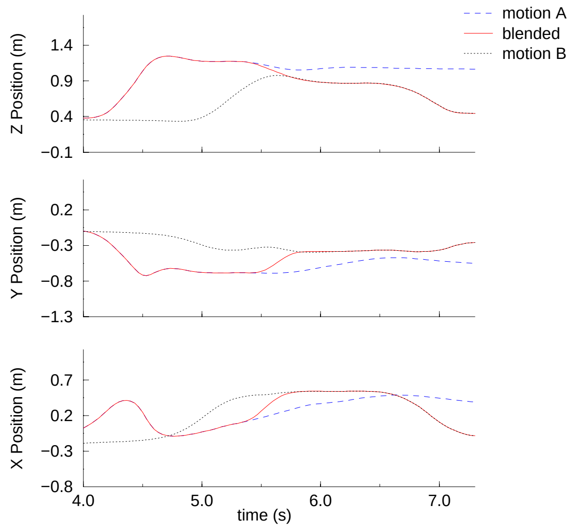

Fig. 5. **Transitions with dynamic filtering** A smooth transition is made between two motion sequences, from motion A to motion B, using a dynamic simulation to filter interpolated joint trajectories. The resulting motion obeys a consistent set of dynamic constraints. The active hand position of the blended and original filtered motions are shown.

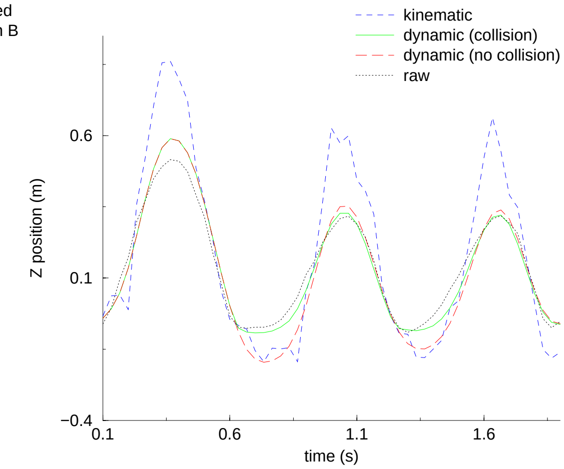

Fig. 6. **Comparison of hand heights in drumming behavior, filtered with and without collision constraints** This graph shows the vertical position of the hand as it drums with contacts at each valley. The raw Cartesian data is taken directly from marker data. The forward kinematic data is calculated from body-orientation data. The dynamic data is the result of dynamic filtering shown with the drum/hand collision constraints on and off. Note, the dynamic model without collisions and the forward kinematic model yield undesirable hand/drum penetrations.

control described here include using dynamic filtering for sub-systems such as arms and legs and motor control for the remainder of the DOFs.

Although constraints that enforce environmental conditions have been implemented with IK solvers, some interactions are more appropriately handled by a dynamic model. Some environmental constraints, such as a hand hitting a drum, are dynamic and IK solutions result in less than appealing motion. In the drum example (see Figure 7), an IK solution could indeed place the hand on the drum at the point of contact and restrict the hand from penetrating the drum. However, the impact and related reaction force would have to be crafted by hand because the model contains no notion of dynamics. A dynamic model can enforce the hand position constraints as well as the dynamic reaction forces corresponding to the drum impact.

Dynamic drum/hand collisions and reactions are added to the system in straightforward manner, based on a collision handler described by O'Brien, Zordan and Hodgins [8]. This handler detects collisions between the vertex in the hand and the polygons of the drum using a bounding-box hierarchy. Reactions are calculated based on penalty parameters and position and velocity errors. Appropriate forces and moments are applied to the dynamic system. The raw motion is filtered as before but as the simulation is integrated, the collision handler corrects constraint violations with reaction forces. As the hand hits the drum, a believable impact is observed. The resulting hand motion closely matches the raw marker data as seen in the graph in figure 6.

These two examples are meant for illustrative purposes, they suggest the potential of a dynamic model in solving complex dynamic constraints. In general, the correct space to solve complex constraints is one in which violation may be detected easily and satisfaction may be performed

Fig. 7. **Drumming motion with physically realistic reaction forces.** Human motion data of a drumming behavior is filtered to animate a simulated character. Constraints between hand and drum surface automatically yield realistic dynamic collision reactions that are impossible to generate with simple kinematic models.

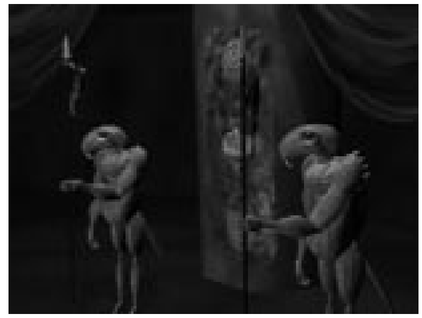

Fig. 8. **Staff-pumping riot motion with task-level constraints.** Human motion data of a staff-pumping behavior is filtered to animate a simulated character. Dynamic constraints to keep the staff upright are imposed explicitly using a simple feedback controller. Secondary motion for the banners is added as a post-process.

in a direct fashion without restricting the motion. For certain dynamic constraints, such as motor-level task and dynamic environmental constraints, a dynamic model affords a natural mapping for satisfying constraints.

VII. Discussion and Conclusions

This paper describes a simple technique for using a dynamic model to filter motion data. The resulting motion is physically plausible for the character being animated but maintains the characteristics of the original motion. The basic filtering system is easily modified to create a transition that has a set of physical laws consistent with the sequences being bridged. Also, two forms of dynamic constraints, namely motor-level task and dynamic contact constraints, are implemented in a straightforward manner to show off the potential of a dynamic model for editing motion.

The graph in figure 9 shows a comparison of hand positions for a crowd-greeting behavior with raw Cartesian data from the hand marker and raw orientation data played through a kinematic skeleton and filtered with the system described. Note that the raw position data is not used in the dynamic filtering process. Rather, orientation data is used as input to the filter. This orientation data is also used in raw form to generate the kinematic hand trajectory shown. Unlike the kinematic profile, the dynamic hand trajectory contains characteristics and overall smoothness similar to the raw hand marker. This is more apparent in the velocity magnitudes seen in figure 10.

The filtered motion resulting from this system has some notable characteristics. In general, the data is smoother than the incoming data. Because the raw data enters the dynamic model as accelerations, the filter is forgiving to spurious data like outliers and small disturbances caused by marker slippage. Further, the dynamic inertial properties minimize the unnatural accelerations allowed by the kinematic model. However, because of the dynamics, extreme values may not be achieved even when desired. This problem can be controlled to an extent by the size of the simulation time step and the control gains, but a more intelligent controller would be needed to correct the problem entirely. Finally, a small time lag is introduced by the filtering system. Although this lag may be a problem when trying to animate tightly timed or choreographed motions, it is sufficiently small and, generally, can be ignored.

This filtering technique is simple but not ideal. The system requires no knowledge about the

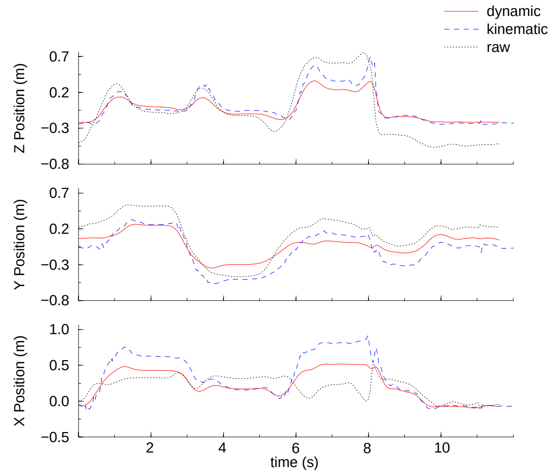

Fig. 9. **Position comparison of hand trajectories for gesturing motion** This graph compares XYZ positions of the active hand during an unobstructed hand gesture. The raw Cartesian data is taken directly from the original hand marker position. The forward kinematic data is calculated from hierarchical body-orientation data. The dynamic data is the result of filtering this orientation data.

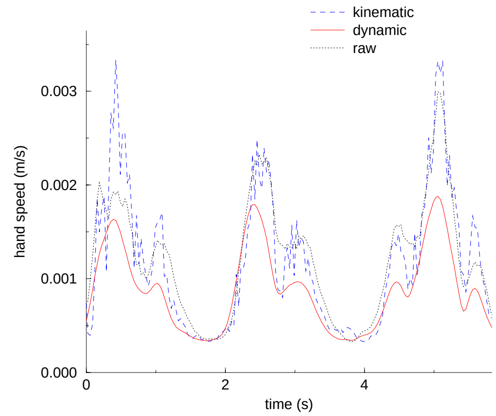

Fig. 10. **Hand speed comparison for gesturing motion** This graph compares the velocity magnitude of the active hand during a hand gesture. Note, the dynamic model uses the same orientation data as the kinematic model but the dynamically filtered motion has velocity profiles more similar to the raw marker data. Also note, the raw data becomes noisy at peak velocities, assumably due to marker slippage during data collection.

motion it is filtering. This is suitable for many cases, but in some scenarios, the system will perform in an unsatisfying manner. More knowledge will make the system more robust. For example, an IK solver that can enforce end effector position is an obvious way to add more knowledge to the filtering process. Also, maintaining balance for a dynamic simulation requires another layer of built-in knowledge in the system. Upper body motion was chosen because the underlying dynamics are inherently stable. However, this system will not generalize to full body motion without a controller that maintains body attitude. Balance controllers have been used in previous systems and implementation is straightforward although conflicts will arise between leg motion tracking and balance control. This level of complexity in dynamic models does not appear in kinematic models.

This paper describes a simple solution for tracking and smoothing human motion data using a dynamic model as well as some interesting methods for modifying motion with complex dynamic constraints. This work does not present a final solution to the question of motion filtering with dynamics. Instead, it addresses the issues of using dynamic models and emphasizes the benefits of dynamics among existing techniques for cleaning raw data and high-level motion editing. However, a variety of exciting and interesting problems related to this work still remain untapped.

Acknowledgments

The author would like to thank Jessica Hodgins and Chris Atkeson for their valuable feedback in this work, James O'Brien and Len Norton for use of their support software, Nancy Pollard and the OOB team for their part in *Alien Occurrence*, Scott Robertson for his efforts in recording the motion data, and Robert Orr, Chad Jenkins, and Scott Myers as motion capture talents.

References

[1] Norman I. Badler, Cary B. Phillips, and Bonnie Lynn Webber. *Simulating Humans: Computer Graphics Animation and Control.* Oxford University Press, New York, 1993.

[2] Bobby Bodenheimer, Charles Rose, Seth Rosenthal, and John Pella. The process of motion capture: Dealing with the data. In *Computer Animation and Simulation '97*, pages 3–18. Eurographics, Springer-Verlag, September 1997.

[3] Armin Bruderlin and Lance Williams. Motion signal processing. In *Proceedings of SIGGRAPH 95*, pages 97–104. ACM SIGGRAPH, August 1995. Held in Los Angeles, California, 6-11 August 1995.

[4] Michael Gleicher. Motion editing with spacetime constraints. In *Proceedings of 1997 Symposium on 3D Graphics*, pages 139–148. ACM SIGGRAPH, April 1997. Held in Providence, Rhode Island, 27 - 30 April 1997.

[5] Michael Gleicher and Peter Litwinowicz. Constraint-based motion adaptation. *The Journal of Visualization and Computer Animation*, 9(2):65–94, 1998.

[6] Jessica K. Hodgins and Nancy S. Pollard. Adapting simulated behaviors for new characters. In *SIGGRAPH 97 Conference Proceedings*, Annual Conference Series on Computer Graphics, pages 153–1–62. ACM SIGGRAPH, Addison Wesley, August 1997. held in Los Angeles, California, 03-08 August 1997.

[7] Jessica K. Hodgins, Wayne L. Wooten, David C. Brogan, and James F. O'Brien. Animating human athletics. In *Proceedings of SIGGRAPH '95*, pages 71–78. ACM SIGGRAPH, August 1995. Held in Los Angeles, California, 6-11 August 1995.

[8] James O'Brien, Victor Zordan, and Jessica Hodgins. Combining active and passive simulations for secondary motion. Technical Report GIT-GVU-97-01, Georgia Institute of Technology, January 1997.

[9] Dinesh Pai. Programming anthropoid walking: Control and simulation. Technical Report 90-1178, Cornell Computer Science, 1990.

[10] Charles Rose, Bobby Bodenheimer, and Michael F. Cohen. Verbs and adverbs. In *CGNA - accepted to appear 11/98*, May 1998.

[11] Charles Rose, Brian Guenter, Bobby Bodenheimer, and Michael F. Cohen. Efficient generation of motion transitions using spacetime constraints. In *Proceedings of SIGGRAPH '96*, pages 147–154. ACM SIGGRAPH, August 1996. Held in New Orleans, Louisiana, 04-09 August 1996.

[12] Karl Sims. Evolving 3d morphology and behavior by competition. In *Artificial Life IV*, pages 28–39, 1994.

[13] Symbolic Dynamics Inc. *SD/Fast User's Manual.* 1990.

[14] Munetoshi Unuma, Ken Anjyo, and Ryozo Takeuchi. Fourier principles for emotion-based human figure animation. In *Proceedings of SIGGRAPH '95*, pages 91–96. ACM SIGGRAPH, August 1995. Held in Los Angeles, California, 6-11 August 1995.

[15] Michiel van de Panne and Eugene Fiume. Sensor-actuator networks. In *Proceedings of SIGGRAPH '93*, pages 335–342. ACM SIGGRAPH, August 1993. Held in Orlando, Florida, 24-29 July 1994.

[16] Michiel van de Panne and Alexis Lamouret. Guided optimization for balanced locomotion. In *Computer Animation and Simulation '95*, pages 165–177. Eurographics, Springer-Verlag, September 1995.

[17] Alan Watt and Mark Watt. *Advanced Animation and Rendering Techniques.* Addison-Wesley, 1994.

[18] Douglas J. Wiley and James K. Hahn. Interpolation synthesis for articulated figure motion. In *Proceedings of IEEE Virtual Reality Annual International Symposium*, pages 156–160, March 1997. Held in Albuquerque, New Mexico, 01-05 March 1997.

[19] Andrew Witkin and Michael Kass. Spacetime constraints. In *Proceedings of SIGGRAPH '88*, pages 159–168. ACM SIGGRAPH, August 1988. Held in Atlanta, Georgia, 1-5 August 1988.

[20] Andrew Witkin and Zoran Popović. Motion warping. In *Proceedings of SIGGRAPH 95*, pages 105–108. ACM SIGGRAPH, August 1995. held in Los Angeles, California, 6-11 August 1995.